

Interaktion

```

*****\   ***\   ***\   *****\   ***\   ***\
\\***\\   ***\   ***\   ***\\   ***\   ***\
***\   ***\   ***\   ***\   *****\   ***\
***\   ***\   ***\   ***\   *****\   ***\   ***\
***\   ***\   ***\   ***\   \\***\   ***\   ***\
***\   ***\   ***\   ***\   ***\   ***\   \\***\
*****\   \\*****\\   *****\\   ***\   ***\
\\*****\\   \\*****\\   \\*****\\   \\   \\

```

```

*****\   *****\
*** ***\   *** ***\
\\ ***\   \\ ***\
***\   ****\
***\   ***\
***\   ***\   ***\
*****\   \\*****\
\\*****\\   \\*****\\

```

September 1986

	Page
ALIEN DODGE. Mel Saunders	2
WHY I FIRED MY SECRETARY	5
WUMPUS. Bruce Joyce	6
TAECOMM NEWS FILE. Tom Evans	12
BLACK HOLE. Allan Weaver	14
EPROM PROGRAMMER AND DESIGN AID. David Parkins	17
INTERAK FLYER. Max Cottrell	22
CONVERSION TABLES	27
BIORYTHMS. Mel Saunders	29
BULLETIN BOARD INFO	31
SOUND-D Mel Saunders	32
DISK SOFTWARE	40
TAPE SOFTWARE	41
SMALL ADS	42
CONTACTS	42

DATA PROTECTION ACT 1985:

Details of the members of the IUGN are held on computer file. Each member may view, alter or destroy any data held on him/her within that file. To obtain a copy of your file please send a stamped addressed envelope to the Editor. Your subsequent wishes regarding that data will be honoured without question.

COPYRIGHT:

All published items remain the copyright of the originators. Members may use published items for their own enjoyment and education but must not describe as their own work or offer for sale any item or part of any item published herein without the express permission of the originator.

ALIEN DODGE
BY MEL SAUNDERS
FOR 64-COL XTAL BASIC

```

10 REM *****
20 REM *'ALIENDODGE' *
30 REM * MEL SAUNDERS 2/84 *
40 REM *****
50 REM
60 REM *****
70 REM * DO YOU HAVE A PSG OR *
80 REM * PCG CARD,THIS PROGRAM *
90 REM * WILL MAKE USE OF BOTH *
100 REM * BUT WILL WORK WITHOUT *
110 REM *****
120 REM
130 REM *****
140 REM * LINES 210-220 DEFINE *
150 REM * MAN AND EXPLOSION *
160 REM *****
170 REM
180 CLS:PRINT@25,12,"PLEASE WAIT"
190 REM:CHANGE CURSOR OTHERWISE
200 REM:ITS A MESS WITH PCG!
210 Q=RD(255)
220 FOR T=&F800 TO &FFFF:POKE T,Q:NEXT T
230 POKE&F800,56,56,16,254,186,186,56,40,40,108
240 POKE&F810,84,149,75,82,58,181,90,60,155,60
250 CLS
260 PRINT@10,4,":INPUT" ENTER NAME PLEASE ";N$:PRINT
270 PRINT@10,6," DO YOU NEED INSTRUCTIONS ";:I=INCH:IF I=89 THEN GOSUB 860
280 PRINT@10,16," WHAT IS YOUR SPEED (0-9) ";:K=INCH-48
290 IF K<0 OR K>9 THEN 280
300 K=K*3
310 PRINT@10,18," WHAT LEVEL (1-9) ";:LL=INCH-48
320 WL=LL*25
330 CLS
340 FOR ALIEN=1 TO WL
350 P=&F000+RD(1536)
360 POKE P,&1C
370 NEXT ALIEN
380 POKE&F5FD,&1E,&1F
390 REM:START POINT.RUN COUNT
400 R=&F002:J=1
410 REM:NO PCG POKE'R' WITH &1D
420 POKE R,&80:POKE R+32,&20
430 FOR T=1 TO 500:NEXT T
440 PRE=R
450 M=KBD
460 IF M=&50 THEN R=R+1:GOTO 490
470 IF M=&51 THEN R=R-1:GOTO 490
480 R=R+64
490 IF PEEK(R)=&1C THEN 590
500 IF PEEK(R)=&1E OR PEEK(R)=&1F THEN 650
510 REM:NO PCG POKE'R' WITH &1D

```

```
520 POKEPRE,32:POKER,&80
530 REM:CONVERT SPEED INPUT
540 S=K/K/K*1000:FORY=1TOS:NEXTY
550 IF R>&F5FF THEN POKER,&20:GOTO810
560 GOTO440
570 REM:CRASH ROUTINE
580 REM:NO PCG POKE'R' WITH &1A
590 POKEPRE,&20:POKER,&81
600 GOSUB1060
610 FORT=1TOS500:NEXT:CLS
620 PRINT@4,11,"you have just had a nasty...ACCIDENT!! when you recover
    would you like....."
630 GOTO 770
640 REM:WIN ROUTINE
650 POKEPRE,&20
660 GOSUB1120
670 FORQ=0T013:OUT&C0,Q:OUT&C1,0:NEXTQ:CLS
680 IOM4,1:IOM5,1
690 PRINT@4,6,"WELL DONE "N$" YOU MADE IT JUST IN TIME TO ESCAPE"
700 IF J=1 THEN R$="RUN"
710 IF J>1 THEN R$="RUNS"
720 PRINT@11,8," IT TOOK YOU"J;R$" AT A SPEED OF";K/3
730 IOM4,0:IOM5,0
740 PRINT@11,10," AND YOUR LEVEL WAS ";LL": "WL
750 IOM4,1:IOM5,1
760 PRINT@11,12," YOUR ASTRO RATING £";LL*(K*K/.35+J):PRINT
770 PRINT@11,14," ANOTHER GAME ? ";:A=INCH:IF A=89 THEN 280
780 CLS:PRINT:PRINT@1,12,"WELL GOODBYE "N$" COME AND PLAY AGAIN SOMETIME
    'STINKER';
790 END
800 REM:NEW RUN
810 R=R-1536:J=J+1
820 IF PEEK(R)=&1C THEN POKER,&20
830 GOTO520
840 REM:INSTRUCTIONS
850 DIM H$(6)
860 CLS: H$(1)=".....ALIEN DODGE
870 H$(2)="*****"
880 H$(3)=" you are at the top left of your scanner 'your' mission is to get
    to your CRAFT IN ORDER TO ESCAPE!"
890 H$(4)=" this you will find at the bottom right of the scanner! But you
    must dodge the aliens that float about in space"
900 H$(5)=" USE THE 'Q'KEY TO TAKE YOU TO THE LEFT AND THE 'P'KEY WILL
    MOVE YOU TO THE RIGHT"
910 H$(6)=" you will be blown to bits if... you hit an alien, BUT just may
    survive for another RUN!"
920 H$(7)="~~~~~ THE GAME WILL START ON ~~~~~
    ~~~~~ ZERO COUNT ~~~~~"
930 FORI=1T07:PRINT
940 FOR J=1TOLEN(H$(I))
950 PRINT MID$(H$(I),J,1);
960 FORT=1T075:NEXTT:NEXTJ:PRINT:NEXTI
970 IOM4,1:IOM5,1
980 FORT=100T00 STEP-1
990 FORQ=1T0100:NEXTQ
```

```
1000 PRINT@26,23,T;:PRINT@30,23,"AND COUNTING";
1010 NEXTT
1020 IOM4,0:IOM5,0
1030 CLS
1040 RETURN
1050 REM:EXPLOSION SOUND
1060 RESTORE 1090:FORT=0TO13:READA
1070 OUT&C0,T:OUT&C1,A
1080 NEXTT
1090 DATA0,0,0,0,0,0,31,7,16,16,16,0,120,0,
1100 RETURN
1110 REM:TAKE-OFF SOUND
1120 RESTORE 1140:FORA=0TO13:READZ:OUT&C0,A:OUT&C1,Z:NEXTA
1130 RESTORE 1140
1140 DATA100,0,0,0,0,0,0,26,20,0,0,0,20,12
1150 REM:TAKE-OFF DISPLAY
1160 FORV=&F5FD TO &F000 STEP-65:POKEV-1,&20,&20:POKEV,&1E,&1F:
FORQ=1TOK/K/K*100:NEXTQ:POKEV,&20,&20:NEXTV
1170 RETURN
```


WHY I FIRED MY SECRETARY

Today was my 40th birthday and I wasn't feeling too hot this morning, but went down to breakfast knowing that my wife would be pleasant and say "Happy Birthday" and probaly have a present for me. She didn't even say "Good Morning" let alone say "Happy Birthday".

I said. "Well that's wives for you, the children will remember". The children came into breakfast and didn't say a word. I was feeling pretty despondent when I started off to work.

As I walked into my office, Wylma said "Good morning boss. Happy Birthday". I felt a little better, as someone had remembered. I worked until noon when Wylma knocked on my door and said, "You know, it's such a beautiful day outside, let's go to lunch, just you and I". I said, "By George that's the greatest thing I have heard all day, let's go".

We went to lunch. We didn't go where we normally go, we went out to the country to a little private place. We had two martinis and enjoyed lunch tremendously. On the way back to the office Wylma said, "Lets go over to my place and I will fix you another martini".

We went to here place, enjoyed another martini and smoked a cigarette, when she said, "Boss, if you don't mind I think I'll go into the bedroom and slip into something more comfortable". I allowed her as I didn't mind at all.

She went into the bedroom and in about six minutes she came out, carrying a big cake, followed by my wife and children singing "Happy Birthday" and there I sat with nothing on but my socks.

WUMPUS
FOR XTAL BASIC
BY BRUCE JOYCE

```

1 CLEAR
10 CLS
20 PRINT@10,6;"";:A$=INCH$
30 IFA$="Y"THEN GOSUB 610
60 CLS:REM**SET UP CAVES
70 DIMS(20,3)
80 FORJ=1TO20
90 FORK=1TO3
100 READS(J,K)
110 NEXTK
120 NEXTJ
130 DATA2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
140 DATA5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
150 DATA10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
160 DATA15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
170 REM**LOCATE ITEMS IN CAVES
180 REM**1-YOU,2-WUMPAS,3&4-PITS,5&6-BATS,7&8&9-ARROWS
190 DIML(R9),M(9)
200 FORJ=1TO9
210 L(J)=RND(20)+1
220 M(J)=L(J)
230 NEXTJ
240 REM**CHECK FOR ERRORS ie.L(1)=L(2)
250 FORJ=1TO9
260 FORK=JTO9
270 IFJ=KTHEN GOTO290
280 IFL(J)=L(K)THEN GOTO210
290 NEXTK
300 NEXTJ
310 REM**SET ARROWS
320 AR=2
330 L=L(1)
340 REM ****
350 AR=2:CLS:PRINT@24,0;"";:M1=0
355 PRINT@40,4;"DO YOU NEED A MAP?";:I$=INCH$:IFI$="Y"THEN L(7)=0:L(8)=0:
    L(9)=0:RESTORE:FORX=1TO236:READD:NEXTX:M1=1:GOSUB 6000
360 REM*HAZARD WARNINGS AND LOCATIONS
370 FORAX=1TO500:NEXTAX:PRINT:GOSUB 1100
380 REM**MOVE OR SHOOT
390 GOSUB 1270
400 ONOGOTO420,460
410 REM***SHOMOT
420 GOSUB 1360
430 IFF=0THEN 370
440 GOTO480
450 REM *** MOVE
460 GOSUB 1860
470 IFF=0THEN 370
480 IFF>0THEN 530
490 REM ***LOSE
500 PRINT@10,21;"HA,HA,HA.YOU LOSE!"

```

```

510 GOTO540
520 REM***WIN
530 PRINT@10,21;"WATCH OUT!!";PRINT@10,22;"*WUMPUS WILL GET YOU";
    PRINT@10,23;"NEXT TIME!!"
540 FORJ=1TO9
550 L(J)=M(J)
560 NEXTJ
570 FORX=1TO2000:NEXTX9:CLS:PRINT@10,10;"SAME SET UP(Y/N)?";
    PRINT@10,12;"PRESS 'Q' TO QUIT"
580 I$=INCH$
590 REM
600 IFI$="N"THENGOTO200
605 IFI$="Y"THENGOTO320:ELSECLS:END
610 REM**INSTRUCTIONS
620 CLS:PRINT@28,0;MUL$(" ",8):PRINT@28,1;" ";PRINT@28,2;MUL$(" ",8)
630 PRINT@0,4;"The wumpus lives in a cave of 20 rooms.each room has
    3 tunnels leading to other rooms.
640 PRINT"(The idea is based on a dodecahedron.)"
650 PRINT"There are several hazards to encounter in the caves.
660 PRINT"PITS:-Two rooms have bottomless) pits in them.
    If you enter them.....
670 PRINT"SUPER BATS:-Two other rooms have super bats in them.
    If you enter then you will be grabbed by a bat and carried off.
680 PRINT"If they dropped you in a pit or on the wumpus then.....
690 PRINT"WUMPUS:-The wumpus is not usually bothered by the hazards as
    he has big sucker feet and is very heavy.
700 PRINT"shhh!He is asleep.The two things that wake him are your going
    into his room or you shooting your arrow.
710 PRINT"If the wumpus wakes up he may move into an adjoining room or
    stay where he is.
720 PRINT"If when he settles you are in that room....wumpus is always hungry
730 PRINT:PRINT"Press any key to continue":X=INCH:FORX=4TO20:
    PRINT@0,X;SPC(64):NEXTX
740 PRINT@0,4;"At each turn you have to decide whether to move or shoot a
    guided arrow.
750 PRINT"MOVING:-You may travel through any tunnel to another room
760 PRINT"ARROWS:-You start with two arrows.there are a further three for
    you to find in the cave.If you run out you lose.
770 PRINT"Each arrow can travel from one to five rooms.You guide the arrow
    by telling the computer the rooms that you want it to go
780 PRINT"through.If there is no tunnel to that room then the arrow moves
    at random to the next room.
790 PRINT"If the arrow hits the wumpus then you win.Beware,because the
    tunnels interlink you could shoot yourself.
800 PRINT"The computer will warn you when you are one room away from a
    hazard.Those bottomless pits are very draughty places.
810 PRINT"You will be offered a map to use.This is really only for those
    who are desperate,try not to use it.
820 PRINT"Use of the map destroys the extra arrows.
830 X=INCH:FORX=4TO20:PRINT@0,X;SPC(64):NEXTX
840 PRINT@0,4;"Please remember to always enter move coordinates* and
    arrow guidance with two digits:-"
850 PRINT"          eg.Move to room 03 not 3
                                arrow to room 09 not 9
1080 I$=INCH$

```

```

1090 RETURN
1100 REM **PRINT LOCATION & HAZARDS
1110 GOSUB3000
1120 FORJ=2TO6
1130 FORK=1TO3
1140 IFS(L(1),K)<>L(J)THENGOTO1210
1150 ONJ-1GOTO1160,1180,1180,1200,1200
1160 CLS:GOSUB4000: PRINT@40,110;"
1170 GOTO1210
1180 PRINT@40,11;"
1190 GOTO1210
1200 PRINT@40,12;"
1210 NEXTK
1220 NEXTJ
1230 PRINT@40,14;"YOU ARE IN ROOM";L(1)
1232 FORJ=7TO9:IFL(1)=L(J)THENPRINT@40,13;"":L(J)=0:A9R=AR+1:GOTO1240
1235 NEXTJ
1240 PRINT@40,15;"TUNNELS LEAD";S(L,1);S(L,2);S(L,3)
1260 RETURN
1270 REM** CHOOSE OPTION
1280 PRINT@40,17;"SHOOT OR MOVE(S/M)?"
1290 I$=INCH$
1300 IFI$<>"S"THEN1330
1310 O=1
1320 RETURN
1330 IFI$<>"M"THENGOTO1280
1340 O=2
1350 RETURN
1360 REM**ARROW ROUTINE
1370 F=0
1380 REM**PATH OF ARROW
1390 L=L(1)
1400 PRINT@40,18;"No.OF ROOMS(1TO5)?"
1410 J9=VAL(INCH$)
1420 IFJ9<1ORJ9>5THENGOTO1400
1430 FORK=1TOJ9
1440 PRINT@40,19;"ROOM £";
1450 P(K)=VAL(INCH$(2)):IFP(K)>20ORP(K)<1THENGOTO1440
1460 IFK<=2THEN1500
1470 IFP(K)<>P(K-2)THENGOTO1500
1480 PRINT@40,20;"TRY ANOTHER ROOM"
1490 GOTO1440
1500 PRINT@100,20;SPC(26):PRINT@100,21;SPC(26):NEXTK
1510 REM** SHOOT ARROW
1520 FORK=1TOJ9
1530 FORK1=1TO3
1540 IFS(L,K1)=P(K)THENGOTO1700
1550 NEXTK1
1560 REM** NO TUNNEL FOR ARROW
1570 L=S(L,RND(3)+1)
1580 GOTO1710
1590 NEXTK
1600 PRINT@40,20,"MISSED"
1610 L=L(1)
1620 REM ** MOVE WUMPUS

```



```

1630 GOSUB1780
1640 REM**ARROW CHECK
1650 AR=AR-1
1660 IFAR>0THENGOTO1680
1670 CLS:PRINT@10,10;"";F=-1
1680 RETURN
1690 REM ** SEE IF YOU GOT SOMEONE
1700 L=P(K)
1710 IFL<>L(2)THENGOTO1750
1720 GOSUB4000:GOSUB4200 :PRINT@10,2 ;""
1730 F=1
1740 RETURN
1750 IFL<>L(1)THENGOTO1590
1760 CLS:PRINT@10, 2;""
1770 F=-1:RETURN
1780 REM** MOVE WUMPUS ROUTINE
1790 K=RND(4)+1
1800 IFK=4THENGOTO1820
1810 L(2)=S(L(2),K)
1820 IFL(2)<>LTHENGOTO1850
1830 PRINT@10,02;"";GOSUB4100
1840 F=-1
1850 IFF>0THEN CLS:IFM1=1THENGOSUB6000
1855 RETURN
1860 REM*MOVE ROUTINE
1870 F=0
1880 PRINT@40,18;"WHERE TO?";
1890 L=VAL(INCH$(2))
1900 IFL<10RL>20THENGOTO1880
1910 FORK=1TO3
1920 REM**CHECK LEGAL MOVE
1930 IFS(L(1),K)=LTHENGOTO1990
1940 NEXTK
1950 IFL=L(1)THEN1990
1960 PRINT@40 ,19;"NOT POSSIBLE"
1970 GOTO1880
1980 REM*CHECK FOR HAZARDS
1990 L(1)=L
2000 REM ** WUMPUS
2010 IFL<^>L(2)THENGOTO2080
2020 PRINT@40,18;"...OOPS! A WUMPUS!!!"
2025 FORX=1TO1000:NEXTX
2030 REM **MOVE WUMPUS
2040 GOSUB1790
2050 IFF=0THEN2080
2060 RETURN
2070 REM** PIT
2080 IFL<>L(3)ANDL<>L(4)THENGOTO2130
2090 CLS:PRINT@10,10;""
2100 F=-1
2110 RETURN
2120 REM*BATS
2130 IFL<>L(5)ANDL<>L(6)THENGOTO2170
2140 PRINT@10,2;""
2150 L=RND(20)+1

```

```
2155 FORX9=1T01000:NEXTX9:PRINT@10,2;SPC(37)
2160 GOTO1990
2170 RETURN
2180 END
3000 FORX=1T020:PRINT@40,X+2;SPC(23):NEXTX:RETURN
4000 RESTORE:FORX=1T060:READW:NEXTX
4010 FORX=5T015
4020 FORX(1=10T025
4030 READX3:IFX3=1THENPRINT@X1,X;CHR$(&7F)
4040 NEXTX1,X
4060 DATA0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,
0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0
4070 DATA0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,
0,0,0,0,0,1,51,1,1,1,1,1,1,1,1,1,1,1,1,1,0
4080 DATA0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,
0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1B,0,0,0
4090 DATA0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1
4095 RETURN
4100 SET31,34:SET32,34:SET39,34:SET40,34
4105 FORZ=1T0 9
4110 SET31,31:SET33,31:SET36,31:SET37,31:SET41,31
4120 SET30,30:SET31,30:SET32,30:SET34,30:SET345,30:SET36,30:SET38,30:
SET40,30:SET41,30
4130 FORX=1T0500:NEXTX:PRINT@15,8;" "
4140 FORX=1T0200:NEXTX:NEXTZ
4145 RETURN
4200 RESET31,34:RESET32,34:RESET39,34:RESET40,34
4210 SET32,34:SET33,34:SET38,34:SET39,34
4220 PRINT@21,12;"<-----<<<"
4230 RETURN
6000 D=1
6010 FORT=1T034
6020 GOSUB6240
6040 IFA=A1THENGOTO6190
6050 C=(B1-B)/(A1-A)
6060 IFA<A1THENLEITS=1:ELSELETS=-1
6070 IFB<B1THENLETC=ABS(C):ELSELETC=ABS(C)*(-1)
6080 LETD=B
6090 IFABS(A-A1)<10THENGOTO6100
6100 IFA>A1THENLETA=A-1:ELSELETA1=A1-1
6110 IFC<1THENLETS1=-1:ELSELETS1=1
6120 FORX=ATOAI9STEPS
6130 FORX1=DTOD+CSTEPS1
6140 SETX,X1
6150 NEXTX1
6160 D=D+C
6170 NEXTX
6180 NEXTT
6185 GOTO6400
6190 IFB>B1THENSWAPB,B1
6200 FORX=BTOB1
6210 SETA,X
6220 NEXTX
6230 NEXTT
6235 GOTO6400
```

```

6240 IFD=1THENREADA,B,A1,B1:D=0:RETURN
6250 A=A1:B=B1
6260 READA1,B1:RETURN
6270 DATA40,4,5,15,10,36,70,36,75,15,40,4
6280 DATA40,10,18,18,20,30,60,30,62,18,40,10
6290 DATA29,14,35,16,45,16,50,22,40,26,30,22,35,16
6300 DATA45,16,51,14,62,18,75,15,70,36,60,30,61,24,50,22,40,26,40,30,20,
    30,10,36,5,15,18,18,19,24,30,22
6400 FORCX=48TO50
6405 FORX1=48TO57
6406 IFX=48ANDX1=48THENX1=49
6410 READA,B
6420 PRINT@A,B;CHR$(X+&80);CHR$(X1+&80)
6425 IFX=50THENRETURN
6430 NEXTX1,X
6440 RETURN
6500 DATA19,21,19,18,25,16,30,15,36,16,35,5,4,5,3,16,9,15,13,16,17,15,21,
    15,24,12,3A0,12,29,8,20,8,9,8,8,12,15,12,19,10

```

Try it NOW

Tascam birth date 1978/03

** UPDATE 17/07/86 **

"Response at last"

We seem to be getting an exchange of dialogue, now that a reply option has been included in the General Message section, plus users helping one another out with various problems. I haven't managed to get the continuous flow to message reads and replies, but as still working on it very hard (honestly). I have been looking at some other BB software, but it's not as user friendly as this package, plus I am just getting to know how to make the changes, so I think I will stick to what I have got for the present, unless of course I run into some big problems.

My disk drive problem "seems" to be solved, after suffering many seek errors over a long period of time, I find that my regular routine maintenance schedules are to blame. I used a silicon lubricant (mass supplied on request) on the main head slides, this turned out to be a bit of a disaster, as it resuspended black dust, and effectively glued the head guides to the slides. Now the rule is NO lubricants to be used at any time, I really am a glutton for punishment!

The hunt for a new auto band scan model is proving very disappointing, lots of rubbish quoted in the adverts, but when you get down to the nitty gritty, they prove next to useless, and very expensive, so will have to wait for a while for accurate cash, and hope for better performance by the manufacturers. I would dearly love to get the higher speeds, but can't justify the cost at present.

My interak passed its V.A.T test when the Customs and Excise carried out the periodic V.A.T inspection, and said they were impressed with the machine and software (of course I wrote it), saying it was ok and I can continue using it.

TAECOMM NEWS FILE

By Tom Evans.

SYSOP for "TOM'S BULLETIN BOARD"

A new Interak bulletin board has started testing daily from the hours of 11pm through 7am the following morning.

FLASH GORDONS BULLETIN BOARD

300 Baud 2 Stop No parity

Dronfield (0246) 410873

Address :-

The Planet Mars

C/O 229 Stonelow Road,

Dronfield,

Derbyshire.

S18 6ER.

System Operators

Flash Gordon,

Dale Arden,

and Professor Zarkov.

Try it NOW

Taecomm birth date: 14/10/85

** UPDATE 17/07/86 **

"Response at last"

We seem to be getting an exchange of dialogue, now that a reply option has been included in the General Message section, plus users helping one another out with various problems, I haven't managed to get the continuous flow to message reads and replies, but am still working on it very hard (honestly!). I have been looking at some other BB software, but its not as user friendly as this package, plus I am just getting to know how to make the changes, so I think I will stick to what I have got for the present, unless of course I run into some big problems.

My disk drive problems "seem" to be solved, after suffering many seek errors over a long period of time, I find that my regular routine maintenance schedules are to blame, I used a silicon lubricant (name supplied on request) on the main head slides, this turned after a few days to something that resembled black Bostic, and effectively glued the head guides to the slides. Now the rule is NO lubricants to be used at any time, I really am a glutton for punishment!

The hunt for a new auto baud scan modem is proving very disappointing, lots of rubbish spouted in the adverts, but when you get down to the nitty gritty, they prove next to useless, and very expensive, so will have to wait for a while to accumulate cash, and hope for better performance by the manufacturers. I would dearly love to get the higher speeds, but can't justify the cost at present.

My Interak passed its V.A.T test when the Customs and Excise carried out the periodic V.A.T inspection, and said they were impressed with the machine and software (of course I wrote it), saying it was o.k and I can continue using it

for the accounts, well I wouldn't fancy going back to the old entry by hand method now anyway (yuk). Do you wanna buy any Accounts software, after that boost I think I will put it on sale.

Precious time has been taken up of late in the design (forced on me) of a replacement power drive control for one of Barbara's knitting machines, I had previously made the drive with direction changes via a relay, micro switches, and a bicycle brake cable for thread colour changes (really clever stuff), but although worked very reliably for 4 years, the old fart next door complains that it is too noisy, now I being a very sensitive person (plus knowing I would never get away with shooting the old sod) decided to re-design the direction changing mechanism (the noisy bit), well the prototype ran succesfully the other day, working on magnetic switches and one SCR, plus a length of "N" gauge railway track for the sliding stops circuitry, it is now really quiet, that is till the old boy next door finds something else to moan about, no pleasing some twits.

See yasTom.

BLACK HOLE

By Allan C Weaver

(For a VDU-K 32 chars x 24 lines)

[Ed - I have no details of what you do to play the game - sorry.]

```

6000 ED 73 E0 60 CD F0 61 2A DE 60 3E 1E 77 3C 23 77
6010 21 48 F0 3E 6F 06 03 77 CD 3A 61 23 CD 3A 61 77
6020 CD 3A 61 CD 3A 61 2B 2B 77 11 12 00 19 77 CD 3A
6030 61 23 23 23 CD 3A 61 77 10 DD 21 00 F0 06 20 3E
6040 7F CD A5 60 21 E0 F2 06 20 CD A5 60 21 1F F0 06
6050 17 77 23 77 C5 06 1F 23 10 FD C1 10 F4 21 61 F0
6060 CD AA 60 CD FC 60 CD FF 61 CD 89 62 FE 4E CA 00
6070 00 FE 4D 20 12 3A E2 60 FE 00 28 06 AF 32 E2 60
6080 18 05 3E 01 32 E2 60 FE 5A 20 12 3A E3 60 FE 00
6090 28 06 AF 32 E3 60 18 05 3E 01 32 E3 60 CD E4 60
60A0 CD FF 61 18 BB 77 23 10 FC C9 3A E2 60 FE 00 28
60B0 16 23 3E 7F BE 2B 28 16 AF 77 23 CD 4D 61 3E 1B
60C0 77 3E 01 32 E2 60 C9 2B 3E 7F BE 23 28 EA AF 77
60D0 2B CD 4D 61 3E 1B 77 AF 32 E2 60 C9 00 FF 8F F2
60E0 00 00 00 00 E5 C5 11 01 00 21 7C 15 CD F7 60 38
60F0 03 2B 18 F8 C1 E1 C9 E5 ED 52 E1 C9 3A E3 60 FE
6100 00 28 1C CD 3A 61 3E 7F BE CD 43 61 28 1C AF 77
6110 CD 3A 61 CD 4D 61 3E 1B 77 3E 01 32 E3 60 C9 CD
6120 43 61 3E 7F BE CD 3A 61 28 E4 AF 77 CD 43 61 CD
6130 4D 61 3E 1B 77 AF 32 E3 60 C9 D5 F5 11 20 00 19
6140 F1 D1 C9 D5 F5 11 20 00 ED 52 F1 D1 C9 3E 6F BE
6150 F5 3E 1E BE 28 6B 3C 23 BE 28 28 65 F1 C0 ED 7B
6160 E0 60 CD 41 62 AF 32 DC 60 CD 6F 61 C3 00 60 E5
6170 21 22 F0 11 96 61 CD BB 61 21 42 F0 11 B1 61 CD
6180 BB 61 C1 CD 89 62 FE 59 C8 FE 4D C8 FE 5A C8 FE
6190 4E CA 00 00 18 ED 59 4F 55 20 46 45 4C 4C 20 49
61A0 4E 54 4F 20 41 20 42 4C 41 43 4B 20 48 4F 4C 45
61B0 0D 41 47 41 49 4E 3F 0D 1A FE 0D C8 77 23 13 18
61C0 F7 ED 7B E0 60 3A DC 60 3C 32 DC 60 CD F1 61 C6
61D0 80 CD E6 61 CD F1 61 CD E6 61 C1 CD 41 62 FE 0F
61E0 D2 63 62 C3 00 60 C5 06 05 CD E4 60 10 FB C1 C9
61F0 AF 11 00 F0 21 00 F3 12 13 CD F7 60 20 F9 C9 E5
6200 2A DE 60 3A DD 60 FE FF 3E 7F 28 07 2B BE 20 23
6210 23 18 07 23 23 BE 20 0B 2B 2B 3A DD 60 2F 32 DD
6220 60 E1 C9 2B 2B AF 77 23 22 DE 60 3E 1E 77 23 3C
6230 77 E1 C9 22 DE 60 3E 1E 77 23 3C 77 23 AF 77 E1
6240 C9 3A DC 60 F5 FE 0A 38 0C DE 0A F5 3E 31 11 5D
6250 F0 CD 61 62 F1 C6 30 11 5E F0 CD 61 62 F1 FE 0F
6260 C9 12 C9 21 82 F0 11 7C 62 CD BB 61 CD 41 62 AF
6270 32 DC 60 21 00 60 E5 21 83 61 E5 C9 59 4F 55 20
6280 48 41 56 45 20 57 4F 4E 0D 06 00 DB 40 C6 80 30
6290 03 47 18 F7 7B B7 DE 20 C9 06

```

ACEY
BY MEL SAUNDERS
FOR 64-COL XTAL BASIC

```

5 CLS:A$="****ACEY**ACEY****"
10 PRINT@24,2,A$
21 PRINT@20,12,"DO YOU NEED INSTRUCTIONS ";:Z=INCH:IFZ=89 THEN GOSUB 2000
110 Q=100:M=100:CLS
111 IFQ>1000 THEN GOTO 3000
120 CLS:PRINT@24,2,A$:PRINT@20,5,"YOU NOW HAVE ";Q
130 PRINT
140 GOTO 260
210 Q=Q+M
220 GOTO111
240 Q=Q-M
250 GOTO 111
260 PRINT@19,7,"HERE ARE YOUR NEXT TWO CARDS  "
270 A=RND(14)+2
280 IFA<2 THEN 270
290 IFA>14 THEN 270
300 B=RND(14)+2
310 IFB<2 THEN 300
320 IFB>14 THEN 300
330 IFA=B THEN 270
350 IFA<11 THEN 400
360 IFA=11 THEN 420
370 IFA=12 THEN 440
380 IFA=13 THEN 460
390 IFA=14 THEN 480
400 PRINT @31,10,A
410 GOTO 500
420 PRINT@30,10,"JACK"
430 GOTO 500
440 PRINT@30,10,"QUEEN"
450 GOTO 500
460 PRINT@30,10,"KING"
470 GOTO 500
480 PRINT@30,10,"ACE"
500 IFB<11 THEN 550
510 IFB=11 THEN 570
520 IFB=12 THEN 590
530 IFB=13 THEN 610
540 IFB=14 THEN 630
550 PRINT@31,12,B:GOTO 650
570 PRINT@30,12,"JACK":GOTO 650
590 PRINT@30,12,"QUEEN"
600 GOTO 650
610 PRINT@30,12,"KING"
620 GOTO 650
630 PRINT@30,12,"ACE"
640 PRINT
650 PRINT
660 PRINT@25,14,:INPUT"WHAT IS YOUR BET ";M
670 IFM<>0 THEN 680
675 CLS:FORT=1TO10:PRINT@4*T,10*T,"CHICKEN!!!!":FORX=1TO200:NEXTX:
PRINT@4*T,10*T,"
":FORX=1TO100:NEXTX,T

```

```
676 PRINT
677 GOTO 111
680 IFM<=Q THEN 730
690 CLS:PRINT@12,12,"SORRY,MY FRIEND BUT YOU BET TOO MUCH"
700 PRINT@14,13,"YOU HAVE ONLY ";Q
710 GOTO 650
730 C=RND(14)+2
740 IFC<2 THEN 730
750 IFC>14 THEN 730
760 IFC<11 THEN 810
770 IFC=11 THEN 830
780 IFC=12 THEN 850
790 IFC=13 THEN 870
800 IFC=14 THEN 890
810 PRINT@31,20,C
820 GOTO 910
830 PRINT"JACK"
840 GOTO 910
850 PRINT"QUEEN"
860 GOTO 910
870 PRINT"KING"
880 GOTO 910
890 PRINT"ACE"
900 PRINT
910 IFC>ATHEN 930
920 GOTO 970
930 IFC>=B THEN 970
950 CLS:PRINT@29,12,"YOU WIN!!!"
951 RESTORE 390:FORX=0TO13:READA:OUT&C0,X:OUT&C1,A:NEXTX
952 DATA95,0,0,0,0,0,0,26,20,0,0,0,100,8
953 FORT=1TO1000:NEXTT
954 FORX=0TO13:OUT&C0,X:OUT&C1,00:NEXTX
955 FORF=1TO2000:NEXTF
960 GOTO 210
970 CLS:PRINT@27,12,"SORRY,YOU LOST "
977 FORT=1TO2000:NEXTT
980 IFM<Q THEN 240
990 PRINT
1000 PRINT
1010 PRINT@16,16,"SORRY, FRIEND BUT YOU BLEW YOUR WAD"
1020 PRINT@20,17,:INPUT"TRY AGAIN (yes or no) ";C$
1030 IFLEFT$(C$,1)="Y" THEN 110
1040 CLS:PRINT@23,12,"O.K HOPE YOU HAD FUN"
1050 END
2000 CLS:PRINT@24,2,A$:PRINT@2,4," ACEY-DUCEY IS PLAYED IN THE FOLLOWING
MANNER"
2010 PRINT:PRINT" THE DEALER (computer) DEALS TWO CARDS FACE UP YOU THEN
HAVE THE OPTION TO GAMBLE OR NOT DEPENDING-"
2020 PRINT" ON WHETHER OR NOT YOU THINK THE THIRD CARD WILL HAVE A VALUE
BETWEEN THE OTHER TWO....."
2030 PRINT:PRINT" IF YOU DON'T THINK IT WILL BE SO YOU CAN PASS JUST BY
ENTERING A 'RETURN' THE MESSAGE IS A BIT UNKIND!"
2040 PRINT@25,22,"PRESS A KEY TO PLAY ";I=INCH:RETURN
3000 CLS:PRINT@3,3," YOU HAVE BROKE THE BANK YOU NOW HAVE" Q "DOLLARS":
FORB=1TO2000:NEXTB:GOTO100
```


EPROM PROGRAMMING AND A DESIGN AND TESTING AID

David M Parkins, Greenbank Electronics.

Nothing stands still in engineering, especially electronic engineering especially computer engineering.

It was clear to me that Interak needed a new EPROM programmer; particularly as large size (8K, 16K, and beyond) EPROMs are readily available for just a few pounds, and not least because new Interak cards (VDU-2K and SBC-1) actually have space for such EPROMs.

The full details of the design (to be called PGM-1) will have to be another story, but briefly the design calls for 5V only operation, power removal from socket for exchange of EPROMs, so called "intelligent" programming, ability to cope with all currently used EPROMs, and no utilisation of the memory space for the PGM-1 design.

This doesn't sound too demanding, but what a nightmare it turned out to be when the design began. For a start there are at least 4 programming algorithms. The easiest is the one with which most "ZYMON" users will be familiar. +25V, the programming voltage, is applied, address and data are set up on the EPROM pins and a single programming pulse of 50 ms is applied. This means that programming proceeds at the rate of about 50 seconds per K. Much faster than earlier methods but very tedious with the biggest EPROMs, for example a 64K EPROM would take nearly an hour to program if this method was used.

Texas Instruments have complicated things by issuing an errata to their earlier data sheets. Now they say 10 ms (maximum 11 ms) should be used, but fail to say how the user is supposed to tell one type of EPROM from another - difficult as they use the same part numbers as before.

More modern EPROMs (eg 2764, 27128) allow an even faster method:

The programming voltage (21 volts now, just to be awkward) is applied, address and data set up, and 1 ms pulses used. After each pulse the data is verified until (we hope) it is correct. Of course this is a much smaller dose of electricity than used previously and some safeguards are insisted on by the manufacturers so that they can guarantee data will be retained for at least 10 years as previously. Firstly programming must be carried out with a higher voltage than the normal +5V on the Vcc supply pins. The voltage used is 6.0V, and at this voltage all the internal thresholds are raised. The idea is that if the device just scrapes through a verification test at 6.0V, it will be fine when Vcc is lowered to its normal level of 5.0V. However this is not the whole story; when using this "algorithm" for EPROM programming, it is vital for long term data storage to use an "overpulse" - a count is kept of however many 1 ms pulses were required, and the count is multiplied by 4 to give the value of the overpulse. If the number of 1 ms pulses reaches 15 without data verifying the overpulse is given anyway and the device failed if that location does not verify after the overpulse. With this method typical programming rates are 1K in 5 seconds or so.

So far so good, but more complications arise as we move on. Other, bigger EPROMs (eg 27256 and 27512) use a method which is the same, but different.

They use a similar method but this time allow up to 25 of the initial 1 ms pulses, failing the device outright if location does not verify before this limit is reached. The overpulse factor is now reduced to 3 times the number of initial pulses required. The required elevated voltage for Vcc during programming is 6.0V as before, but there is another change in the programming voltage Vpp, which is now specified to be 12.5V. This algorithm is called the "intelligent programming algorithm", which is why we all want to use it. Most modern designs of EPROM programmer allow for this, and a wide choice of Vpp voltages as well, thinking they've cracked it. Using the intelligent algorithm reduces typical programming rates to 1K programmed in about 4 seconds.

Something very new however, and causing even more consternation to the poor designer, is to be found on an Intel data sheet for a production EPROM called the 27256P. As well as allowing the Intelligent algorithm just discussed they introduce a new algorithm called the "Quick Pulse" algorithm. This requires a new Vcc of 6.25 Volts, a new Vpp of 12.75 Volts (both very accurately set), and the use of 100 microsecond pulses as initial pulses. Up to 25 of these may be applied until the data verifies, and the device failed if a location does not verify after that, but no over-pulses are required. Typical programming rates, excluding the overhead of the programmer itself, are 1K programmed in a tenth of a second! You will appreciate that this is about 500 times faster than the rate I mentioned when I began this discussion.

It is obvious that our design has to accommodate the latest types if we are to hold our head up high in the computer world, especially as most other programmer designs have started wrong footed thinking it is only Vpp that is subject to numerous variations, little did they know Vcc was going to be just as volatile in the end.

These weird algorithms and voltages are only one part of the nightmare; another frightener for the poor programmer designer is the number of same but different pinouts. Some EPROMs are in 24-pin packages, some in 28-pin, some have an active low signal for programming, some active high. Addresses and control pins wander about from device to device, and generally things can get pretty messy. Many cowardly designers limit themselves to the "27 series" of EPROMs because they are mostly the same; when they list the number of devices allowed the list looks so comprehensive that the casual user thinks everything is there, but if we're to do the job properly we should also try and include the "25 series" as well. A sticky one at the top of the range is the 27512, because when the 64K EPROM (which is what this is) is reached the 28-pin package is bursting at the seams and it is one of the messier ones to program. Some designers allow for full software selection of "all EPROM types" (except the exceptions of course!) My goal is to accommodate even the awkward ones, without resorting to changing on-board links and the like.

Even though EPROMs are laughingly claimed to be upward compatible, (ie new ones are supposed to fit in old sockets with only the most minor of changes), I discovered that there are 3 distinct places where an evil Vpp voltage can be applied, and 2 places for Vcc, with numerous cases of vagrancy as far as other signals are concerned.

You will see in the data sheets for the first few 28-pin EPROMs great attention is drawn to the fact that programming is carried out by means of TTL level pulses - no need to pulse that nasty old Vpp high voltage line. (In the bad old days this was quite normal, our geriatric Interak clientele will

remember the Kemitron EPROMs type 5204s (0.5K!) which had to be programmed with a pulsed -50 Volt supply, as used by Baron Frankenstein for the manufacture of similar monsters to our early systems.) But the data sheets go strangely silent on this point when we get to the latest 27512 EPROMs. Because of the great shortage of pins in the 28 pin package for this capacity of memory device, Vpp is to be found on the same pin as the output enable line. To read the device, even for verification during programming, the output enable line must be brought low, but it has to be back up at Vpp before another programming pulse can be applied. The needle of a voltmeter connected to that line would move faster than a fiddler's elbow, and if this voltage was connected via relays the resulting death rattle during programming would scare the wits out of even Baron F. I have seen one or two designs which did make the Vpp connection in this way, yet claimed to be able to program 27512s, but I can't poke too much fun at them, my Mk 1 design did exactly the same until I realised what a clanger I'd dropped - at least nobody else got to hear about it except you, and you deserve a laugh as a reward for reading this far.

Anyway, I've probably said enough to indicate that it is an extreme test of mental stability to undertake the design of a universal EPROM programmer, and of course the software to drive it. I have undertaken both and now have to move on quickly before the nice men with the white coats in the van with leather windows come to take me home.

Chasing pulses voltages and universal algorithms around a 28-pin socket whilst everything was happening at once soon proved to be no fun at all so I devised a simple piece of test apparatus to make the job much easier for me. It helped me so much I at first thought I would keep the idea as a trade secret, but hearing that many of you are conducting experiments along similar lines, I decided to share my genius with the rest of you, to save you time and trouble too.

What is this world shattering invention? I am sorry for the anti-climax but all it is is a set of LEDs (light emitting diodes) connected to each signal line on a 28-pin header which plugs into the socket of the EPROM programmer. Much careful work has to be done in the software design to ensure correct operation and avoid damage. It is a computer programmer's dream, because everything depends on everything else and what has gone before. Vpp must never be applied if Vcc is absent, there must be a specified delay after address and data set up before programming, and after programming and before verification, and so on and so on. I had the great joy with one of my experiments in successfully programming a handful of EPROMs supplied to customers who had returned them as being "defective - unable to program"; on their poxy machines may be, not on my Super Whizzo! And how did I do it? with my little LEDs.

The accompanying diagrams give the circuit, which is very simple. To make it look more complicated than in fact it is, green LEDs have been used for the data lines, red for the rest, but you can use any colours you fancy. LEDs are diodes and conduct current very freely in the forward biased direction, like any diode, and so need resistors to limit this current to a safe value. A value of 1k in series with a red LED gives a clear illumination when logic levels are applied but keeps current to less than 25 mA even if a Vpp (programming voltage) of 25 Volts is applied. Vpp can be applied to 3 pins on this header, pins 1, 22, 23, and in these positions the "1k" comprises a pair of 2k2 resistors wired in parallel. This is because the power dissipated in the resistors when 25 volts is applied is around 0.5 Watts, which of course is

too much for a single 0.25 Watt resistor (0.25 Watt resistors have to be used in my design because space on the 28-pin header is so cramped). There is no such restriction on the eight green LEDs connected to the data signals, and 470 ohm resistors have been used here because green LEDs typically require smaller resistors than would be used for red LEDs to achieve the same brilliance.

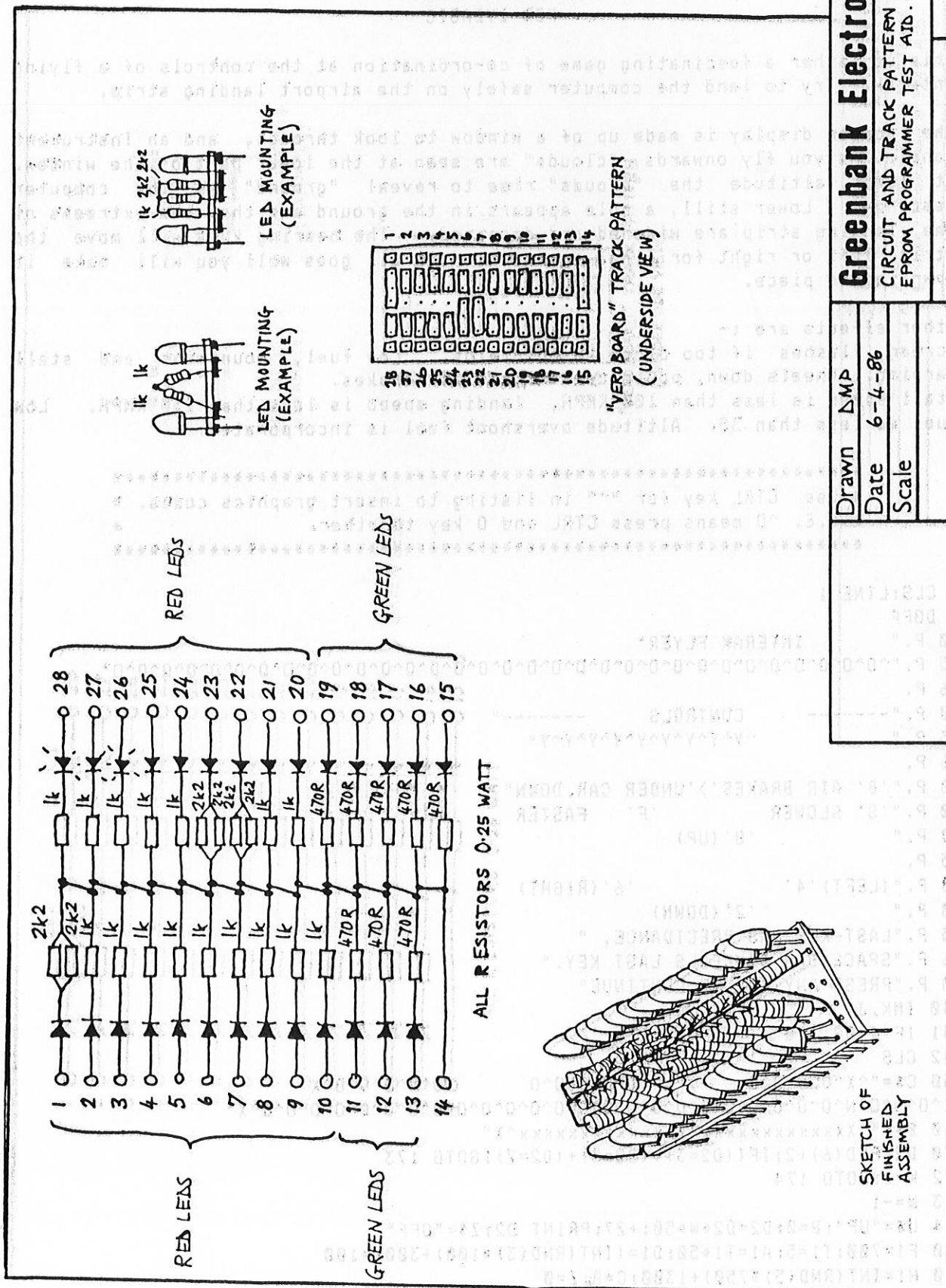
In use the control program is taken slowly through its paces, and the operation of the LEDs observed at each stage. It is easy to confirm that the socket is "dead" because all the lights are off, and very quick and easy to trace the power up sequence (if any) when Vcc and Vpp are applied and then raised to higher voltages during the programming phase, because of course the higher voltage makes the LED glow more brightly. The various modes of operation during programming involve different logic levels on various pins (chip enable, output enable, the PGM line, etc.) and these can be seen clearly on the appropriate LEDs. Even when the program is run in real time the LEDs can provide information, for example those who know their binary counting sequences can see them ascending on the address line LEDs as "programming" proceeds. Once the program has been checked with the dummy EPROM, ie the arrangement of LEDs described, and some pulsewidth measurements taken of the critical timing pin(s) it is fairly safe to plug in a real EPROM and watch it burn!

Construction is easier to execute than to describe. I used a small piece of Veroboard with copper strips, and cut grooves across each track with a scalpel wherever I wanted to break the connection. The LEDs I used were the smallest I had (3 mm) so I had to file them down so that they could be stacked on 0.1" centres. The anode (long) lead of each LED was stiff enough and long enough to make each lead of the finished dummy EPROM. The resistors were all soldered by one end in position on the board, in series with the cathode (short) lead of the appropriate LEDs, and all the free ends gathered together and connected in common to pin 14 of the finished job, the 0V pin, see sketch.

The building of the full "PGM-1" design in its present form is only recommended to advanced constructors, as it is so full of rubbish (for the reasons already discussed at length) that it takes a couple of DIP-1 boards mounted piggy-back fashion to accommodate it in its entirety. However anyone who wants it can have a copy of the circuit diagram for some charge (probably couple of pounds - apply to Greenbank Electronics). Software to drive the board is only half baked at the time of writing but can be supplied on an Interak standard 3.5" disk to run under CP/M as a ".COM" file; (price and specification of software to be determined).

Needless to say a printed circuit board is to be laid out for the PGM-1 design, but it is going to be a real squeeze to fit it on one board, but we'll try.

CIRCUIT FOR THE EPROM PROGRAMMER TEST AID



Greenbank Electronics	
CIRCUIT AND TRACK PATTERN FOR EPROM PROGRAMMER TEST AID.	
Drawn ΔMP	
Date 6-4-86	
Scale	

**INTERAK FLYER
BY MAX COTTRELL
FOR ZYBASIC**

This is rather a fascinating game of co-ordination at the controls of a flying Interak. Try to land the computer safely on the airport landing strip.

The screen display is made up of a window to look through, and an Instrument panel. As you fly onwards, "clouds" are seen at the lower part of the window. At lower altitude the "clouds" rise to reveal "ground" as the computer descends. Lower still, a hole appears in the ground and then the extremes of the landing strip are widened and deepened. The bearing keys will move the strip left or right for final approach. If all goes well you will make it down in one piece.

Other effects are :-

Screen flashes if too close to mountains. Low fuel, mountains and stall warning. Wheels down, power control and air brakes.

Stall speed is less than 100 KMPH, landing speed is less than 250 KMPH. Low fuel is less than 30. Altitude overshoot feel is incorporated.

```
*****
* Use CTRL key for "^" in listing to insert graphics codes. *
* I.E. ^O means press CTRL and O key together. *
*****
```

[illegible]

Page 23

```

599 GOTO 603
600 IF D2 >100 GOSUB 1100
601 IF D2 <-100 GOSUB 1100
603 IFA1<100 GOSUB1200
605 IFH1<1000 GOSUB1300:GOTO615
610 FORA=1TO 3:LINE 22:P."^0
P."^0
615 GOSUB 4000
616 IF D1<0 GOTO 619
617 GOTO 680
619 IF H1> 0 GOTO 621
620 GOTO 680
621 IF ABS(D2) <100 GOTO 5400
680 GOTO 450
1000 FOR E=1 TO 6
1010 FOR E=1 TO 6
1020 LINE 22:P."
1030 LINE 23:P."
NEXT E:Z=Z-5
1040 IF F1<0 LET T1=0
1050 LINE 1:RETURN
1100 IF D1<=0 GOTO 5300
1110 LINE3:P.C$
1120 FOR F=1 TO6: B=1
1130 LINE 22:P."^0",,,,,,,,,,"
1140 LINE 23:P."^0",,,,,,,,,,"
NEXT A:NEXT F:Z=Z-10
1150 RETURN
1200 FOR G=1TO6
1220 LINE22:P."^0",TAB(21)"
1230 LINE 23:P."^0",TAB(21)"
1240 H1=H1-35:T1=0:A1=A1+28:IF H1<=0 GOTO5200
1250 RETURN
1300 J=EF0E0:IF D1>=10000 GOTO1400
1315 IF ABS(D2) >100 GOTO 1500
1320 IF D1>=8000 GOTO1326
1321 GOTO 1330
1326 IF ABS(D2) <=100 GOTO 1600
1330 IFD1>=5000 GOTO 1336
1331 GOTO 1340
1336 IF ABS(D2) <=100 GOTO1700
1340 IF D1>=3000 GOTO 1346
1341 GOTO 1350
1346 IF ABS(D2)<=100 GOTO 1800
1350 IFD1>1000 GOTO 1356
1351 GOTO 1360
1356 IF ABS(D2)<=100 GOTO1900
1360 IF D1<=1000 GOTO 1370
1370 IF ABS(D2)<=100 GOTO 2000
1400 LINE 8:FOR A=1 TO 4: P.B$:NEXT A
1410 IF H1 <=0 G. 5000
1420 GOTO3000
1500 LINE 3:P.C$:
1520 GOTO 3000
1600 Z=Z+5

```

FUEL",,,,,,,,,,,,,,,,,,"^0":FOR A=1 TO 40:NEXT A
 FUEL",,,,,,,,,,,,,,,,,,"^0":FOR A=1 TO 40:NEXT A:
 AINS",,,,,,,,,,"^0:FOR A=1TO 40:NEXT A
 AINS",,,,,,,,,,"^0:FOR A =1 TO 40:
 TALL^0":FOR A=1TO40:NEXT A
 STALL^0":FOR A=1 TO 40:NEXT A: NEXT G:Z=Z-10

Produced on an INTERAK

```
5045 IF U$<>"UP"GOTO5047
5046 GOTO 5050
5047 IF A1<200 P. INT(RND(2)*10),"PEOPLE WERE INJURED":GOTO 6000
5050 IFU$<>"LO" GOTO 5053
5051 IF A1 >200 GOTO 5053
5052 GOTO5060
5053 P." THERE WERE NO SURVIVORS":Z=Z-100:GOTO 6000
5060 FOR A=1 TO 1000:NEXT A: P."^\"
5070 CLS:IF ABS(D2)<20 P."YOU LANDED ON THE RUNWAY",D2,"METRES"
5075 IF ABS(D2)<20 P."OFF CENTER &",D1,"METRES FROM":Z=Z+30
5080 IF ABS(D2)>20 P."YOU JUST MISSED THE RUNWAY,YOU"
5085 IF ABS(D2)>20 P.D2,"METRES OFF CENTER &",D1,"METRES"
5087 IF ABS(D2)>20 P."FROM THE":Z=Z+10
5090 P."AIRPORT.YOU LANDED AT",A1,"KMH"
5100 IF U$<>"LO" P."YOUR UNDER CARRIAGE WAS NOT DOWN"
5105 IF U$<>"LO" P."THE PLANE WAS A WRITE OFF":Z=Z-50
5110 IF A1<250 GOTO6000
5120 D=INT((A1-250)/20):FOR E=1 TO D:F=INT(RND(3)*10)
5130 IF F=7 P."YOUR LANDING SPEED WAS TO GREAT"
5140 IF F=7 P."THE PLANE BROKE UP":Z=Z-80:GOTO6000
5150 GOTO6000
5200 P."^OYOUR PLANE STALLED & CRASHED "
5210 P."ON ITS NOSE,NO SURVIVORS "
5220 P."^OYOU WERE",D2,"METRES OFF COURSE"
5230 P."^OAND",D1,"METRES FROM THE AIRPORT":Z=Z-150:GOTO6000
5300 PAGE:CLS:P."^OYOU SMASHED INTO A MOUNTAIN ":P."AT",A1,"KMH.":Z=Z-200
5320 P."^OYOU WERE",D2,"METRES OFF":P."COURSE." :GOTO 6000
5400 PAGE:CLS:P."^OYOU FLEW OVER THE AIRPORT ":Z=Z-60
6000 P."*****"
6001 P."*****"
6060 Z=Z+INT((200/(ABS(D1)+1))+(150/(ABS(D2)+2))+(F1/15)+(30-(H1/150)))
6065 IF Z<0 LET Z=0
6070 P."YOU SCORED",Z,"POINTS.":IF Z>Q LET Q=Z
6075 P."HI-SCORE:",Q
6077 P."DO YOU WANT ANOTHER GO?"
6078 INK.J
6079 IF J=£59 G. 8
6080 IF J=£4E GOTO 7000
6090 GOTO 6078
7000 STOP
```

CONVERSION TABLES

Centimeters (cm)	Inches (in)	Inches (in)	Centimeters (cm)	Centegrade (C)	Farenhight (F)
1	0.394	1	2.54	0	32
2	0.787	2	5.08	5	40
3	1.181	3	7.62	10	50
4	1.575	4	10.16	15	60
5	1.969	5	12.7	20	70
6	2.362	6	15.24	25	75
7	2.756	7	17.78	30	85
8	3.15	8	20.32	35	95
9	3.543	9	22.86	40	105
10	3.937	10	25.4	60	140
20	7.874	20	50.8	80	175
30	11.811	30	76.2	100	212
40	15.748	40	101.6		
50	19.685	50	127.0		

Kilograms (kg)	Pounds (lb)	Pounds (lb)	Kilograms (kg)	Km/h	Mph
1	2.205	1	0.454	32	20
2	4.409	2	0.907	48	30
3	6.614	3	1.361	64	40
4	8.819	4	1.814	80	50
5	11.023	5	2.268	96	60
6	13.228	6	2.722	112	70
7	15.432	7	3.175	128	80
8	17.637	8	3.629	144	90
9	19.842	9	4.082	160	100
10	22.046	10	4.536		
20	44.092	20	9.072		
30	66.139	30	13.608		
40	88.185	40	18.144		
50	110.231	50	22.68		

Kilometers (km)	Miles (mi)	Miles (mi)	Kilometers (km)	kg/sq cm	lb/sq in
1	0.621	1	1.609	1.41	20
2	1.243	2	3.219	1.55	22
3	1.864	3	4.828	1.69	24
4	2.485	4	6.437	1.83	26
5	3.107	5	8.047	1.97	28
6	3.728	6	9.656	2.11	30
7	4.350	7	11.265	2.25	32
8	4.971	8	12.875	2.39	34
9	5.592	9	14.484		
10	6.214	10	16.093		
20	12.427	20	32.187		
30	18.641	30	48.28		
40	24.855	40	64.374		
50	31.069	50	80.467		

Tonnes (t)	UK tons (T)	UK tons (T)	Tonnes (t)
1	0.984	1	1.016
2	1.968	2	2.032
3	2.953	3	3.048
4	3.937	4	4.064
5	4.921	5	5.08
6	5.905	6	6.096
7	6.889	7	7.112
8	7.874	8	8.128
9	8.858	9	9.144
10	9.842	10	10.161
20	19.684	20	20.321
30	29.526	30	30.481
40	39.368	40	40.642
50	49.21	50	50.802

Hectares (ha)	Acres (acres)	Acres (acres)	Hectares (ha)
1	2.471	1	0.405
2	4.942	2	0.809
3	7.413	3	1.214
4	9.884	4	1.619
5	12.355	5	2.023
6	14.826	6	2.428
7	17.297	7	2.833
8	19.769	8	3.237
9	22.24	9	3.642
10	24.711	10	4.047
20	49.421	20	8.094
30	74.132	30	12.14
40	98.842	40	16.187
50 ...	123.553	50	20.234

Litres (l)	Gallons (gals)	Gallons (gals)	Litres (l)
1	0.22	1	4.546
2	0.44	2	9.092
3	0.66	3	13.638
4	0.88	4	18.184
5	1.1	5	22.73
6	1.32	6	27.276
7	1.54	7	31.822
8	1.76	8	36.368
9	1.98	9	40.914
10	2.2	10	45.46
20	4.399	20	90.919
30	6.599	30	136.379
40	8.799	40	181.839
50 ...	10.998	50	227.298

Paper (Ax)	Milimeters	Inches
A0 ..	841 x 1189	33.1 x 46.8
A1 ..	594 x 841	23.4 x 33.1
A2 ..	420 x 594	16.5 x 23.4
A3 ..	297 x 420	11.7 x 16.5
A4 ..	210 x 297	8.3 x 11.7
A5 ..	148 x 210	5.8 x 8.3
A6 ..	105 x 148	4.1 x 5.8
A7 ..	74 x 105	2.9 x 4.1
(Note:- The length becomes the width at each step up)		

Men's Suits and Coats		
British	American	Continental
36	36	46
38	38	48
40	40	50
42	42	52
44	44	54
46	46	56
48	48	58

Men's Shirts		
British	American	Continental
14	14	36
14.5	14.5	37
15	15	38
15.5	15.5	39/40
16	16	41
16.5	16.5	42
17	17	43

Men's Shoes		
British	American	Continental
7	8	41
7.5	8.5	42
8.5	9.5	43
9.5	10.5	44
10.5	11.5	45
11	12	46

Men's Socks		
British	American	Continental
9.5	9.5	39
10	10	40
10.5	10.5	41
11	11	42
11.5	11.5	43
12	12	44

BIORYTHMS
BY MEL SAUNDERS
FOR 64-COL XTAL BASIC

```

10 CLS:IOM4,0:IOM5,0:FMT0,0
20 B$="" ~~~~~ BIORYTHMS ~~~~~
30 PRINTB$:PRINT:PRINT
40 PRINT" What are BIORYTHMS it's a new science concerned with life..
cycles,and how from birth we move thro'periods "
50 PRINT:PRINT" of High peaks of feeling good.. thinking good,to troughs of
doubts and fatigue and even";
60 PRINT" our emotional state is on the change constantly"
70 PRINT:PRINT" ENTER DATES AS REQUESTED THE GRAPH WILL DISPLAY THE AVERAGE
BIORYTHM FOR THE DATE";
80 PRINT" ENTERED AND THE NEXT 31 DAYS"
90 PRINT@3,20,"PLEASE ENTER NAME AND RETURN (press any key to repeat!)"
95 PRINT@3,22,:INPUT"NAME PLEASE ";N$
100 CLS
110 PRINT:PRINTB$:PRINT
120 PRINT@0,4," "
130 PRINT@0,4,:INPUT"DATE OF BIRTH ";FD
140 PRINT@14,4," "
150 PRINT@0,4,:INPUT"MONTH OF BIRTH ";FM
160 PRINT@14,4," "
170 PRINT@0,4,:INPUT"YEAR OF BIRTH ";FY
180 PRINT@14,4," "
190 PRINT@40,4,"BORN ON- ";FD;"/";FM;"/";FY
200 PRINT@0,4,:INPUT"TODAYS DATE ";BD
210 PRINT@11,4," "
220 PRINT@0,4,:INPUT"WHICH MONTH ";BM
230 PRINT@11,4," "
240 PRINT@0,4,:INPUT"WHICH YEAR ";BY
250 PRINT@0,4," "
255 N$="** BIORYTHM FOR "+N$+"**"
260 PRINT@5,4;N$
270 PRINT@40,6,"BIO-DAY? ";BD;"/";BM;"/";BY
273 REM:ROUGHLY CALCULATE
274 REM:DAYS OF LIFE
275 PRINT@40,9,"DAYS OF LIFE- "INT(365.25*(BY-FY)+(30*(BM-FM)+(BD-FD)))
280 S=0
290 D=FD:M=FM:Y=FY
300 GOSUB 570
305 AD=-X
310 FORZ=FY TO BY
320 IF Z/4-INT(Z/4)=0 AND Z/100-INT(Z/100)<>0 AND FY<>BY THEN S=S+1
330 NEXTZ
340 AD=AD+S+365*(BY-FY)
350 D=BD:M=BM:Y=BY
360 GOSUB 570
365 AD=AD+X+(M>2 AND Y/4-INT(Y/4)=0 AND Y/100-INT(Y/100)<>0)
370 GOSUB 730
390 FMT2,2
400 PRINT@3,6,"PHYSICAL";TAB(30,46)(99*A+.5)
410 PRINT@3,7,"EMOTIONAL";TAB(30)(99*B+.5)
420 PRINT@3,8,"INTELLECTUAL";TAB(30)(99*C+.5)
430 PRINT@3,9,"AVERAGE";TAB(30)(99*D+.5)

```

```

440 FORP=3TO 24STEP2
450 SET0,P
460 NEXTP
470 FORP=0TO127 STEP2
480 SETP,13
490 NEXTP
500 FORG=ADTO AD+30
510 GOSUB 740
515 SET4*(G-AD)+2,INT(10*D+.5)+11
520 SET4*(G-AD)+1,INT(10*D+.5)+11
525 SET4*(G-AD),INT(10*D+.5)+11
530 NEXTG
540 PRINT@1,12,"GOOD DAYS":PRINT@1,22,"BAD DAYS"
550 A$=INCH$:IFA$="" THEN 550
560 RUN
570 X=D
580 IF M=2 THEN X=31+X
590 IF M=3 THEN X=59+X
600 IF M=4 THEN X=90+X
610 IF M=4 THEN X=120+X
620 IF M=5 THEN X=151+X
630 IF M=6 THEN X=181+X
640 IF M=5 THEN X=120+X
650 IF M=6 THEN X=151+X
660 IF M=7 THEN X=181+X
670 IF M=8 THEN X=213+X
680 IF M=9 THEN X=242+X
690 IF M=10 THEN X=273+X
700 IF M=11 THEN X=303+X
710 IF M=12 THEN X=334+X
720 RETURN
730 G=AD
740 A=SIN(2*PI*G/23):B=SIN(2*PI*G/28):C=SIN(2*PI*G/33)
750 D=(A+B+C)/3
760 RETURN
765 REM
770 REM *****
775 REM * BIORHYTHMS BY *
780 REM * MEL SAUNDERS 4/84*
790 REM *****

```

BULLETIN BOARD DATA

TOMS BULLETIN BOARD

Access : Free
 On-line times : 1900 to 0700 mon-sat and all day Sunday.
 Transmission : 300 Baud, 2 Stop bits no parity.
 Data line : 01-573-8822
 Speech line : 01-561-2639 for help and advice
 Sysop : Tom Evans
 Address : 129 Cranborne Waye, Hayes, Middleswx.
 Post code : UB4 0HR

FLASH GORDON'S BULLETIN BOARD

Access : Free
 On-line times : 2300 to 0700 daily.
 Transmission : 300 Baud, 2 Stop bits no parity.
 Data line : 0246-410873
 Speech line :
 Sysops : Flash Gordon, Dale Arden and Professor Zarkov.
 Address : The Planet Mars, C/O 229 Stonelow Road, Dronfield, Derbyshire,
 Post code : S18 6ER.

M&M ELECTRONICS

Bulletin board Software and services to the Interak computer.
 M & M Electronics, 8 Ayre View, Bride, Isle of man.

SOUND-D
FOR XTAL BASIC
BY MEL SAUNDERS

```

10 A$="~~~~~SOUND~DEVELOPEMENT~~~~~"
20 A1$=""
30 B$="~~~~~THE AY-3-8910 REGISTERS~~~~~"
40 C$="~~~~~CLOCKS AND FREQUENCYS~~~~~"
50 D$=MUL$(" ",255)
60 AB$=MUL$(" ",33):A2$=MUL$(" ",36)
70 DIMS(20),T(20)
80 GOTO130
90 PRINT@15,23,"press SPACE to continue ";
100 A=INCH:IF A<>20 THEN 90
110 CLS:PRINTA1$:PRINTA$:PRINTA1$
120 RETURN
130 CLS:PRINT:PRINTA1$:PRINTA$:PRINTA1$
140 PRINT:PRINT:PRINT:PRINT
150 PRINT"THIS PROGRAM WILL HELP YOU USE THE P.S.G CARD TO CREATE YOUR
    OWN SOUNDS AND ALSO UNDERSTAND THE WORKINGS OF-
160 PRINT" THE AY-3-8910 CHIP AND THE INTERAK CARD ":PRINT
170 PRINT" The AY-3-8910 chip has 16 Registers (mostly 8 bit) that have
    control over 3 sound channels, noise, envelope ";
180 PRINT"time and shape.."
190 PRINT:PRINT" There is also input/output by way of 4 8-bit ports.
    that can be very usefull for joystick use or even";
200 PRINT" ROBOT CONTROL!"
210 GOSUB90
220 PRINT:PRINTB$:PRINT
230 PRINT"          R-0 8-BIT FINE TUNE CHANNEL A":PRINT
240 PRINT"          R-1 4-BIT COARSE TUNE CHANNEL A":PRINT
250 PRINT"          R-2 8-BIT FINE TUNE CHANNEL B":PRINT
260 PRINT"          R-3 4-BIT COARSE TUNE CHANNEL B":PRINT
270 PRINT"          R-4 8-BIT FINE TUNE CHANNEL C":PRINT
280 PRINT"          R-5 4-BIT COARSE TUNE CHANNEL C":PRINT
290 PRINT"          R-6 5-BIT NOISE TUNE CHANNEL":PRINT
300 PRINT"          R-7 ENABLE:note active low":PRINT
310 GOSUB90
320 PRINT:PRINTB$:PRINT:PRINT:PRINT
330 PRINT"          R-8 5-BIT CHANNEL A VOLUME":PRINT
340 PRINT"          R-9 5-BIT CHANNEL B VOLUME":PRINT
350 PRINT"          R-10 5-BIT CHANNEL C VOLUME":PRINT
360 PRINT"          R-11 8-BIT FINE TUNE ENVELOPE":PRINT
370 PRINT"          R-12 8-BIT COARSE TUNE ENVELOPE":PRINT
380 PRINT"          R-13 4-BIT ENVELOPE SHAPE/CYCLE":PRINT
390 PRINT"          R-14/15 8-BIT I/O PORTS A AND B"
400 GOSUB 90
410 PRINT:PRINT:PRINT:PRINTC$:PRINT:PRINT
420 PRINT"In most cases the frequency is obtained by firstly dividing
    the clock frequency by 16 then"
430 PRINT:PRINT" by the value in the register(s) this value is a period
    value ie-the lower the value the higher the tone";
440 PRINT"..and vice-versa"
450 PRINT:PRINT" IT IS POSSIBLE TO ADD A 74LS393 TO PROVIDE CLOCK FREQUENCYS
    OF 4MHz TO 15.625KHz ADWDING A VERY";
460 PRINT" WIDE RANGE OF SOUNDS AND TONES":PRINT

```



```

470 PRINT " What is a short sharp crack at 4MHz is a low long bong at
      250KHz the range is very wide almost endless"
480 GOSUB 90
490 PRINT:PRINTC$:PRINT
500 PRINT " TP=fine tone period value      CT=coarse tone period value"
510 PRINT " Fclock=system clock frequency  FT=desired tone frequency":PRINT:
      PRINT "      a bit of maths will help-"
520 PRINT:PRINT "      ie.FT=1KHz  Fclock=2MHz "
530 PRINT:PRINT "      2MHz/16/1000=125 TP.VALUE "
540 PRINT "      or 2MHz/16/125=1000 or 1KHz"
550 PRINT:PRINT " A 4MHz clock will produce a tone range of 60Hz to 250KHz
      while a 1MHz clock would produce a tone range of";
560 PRINT " 15Hz to 62.5KHz"
570 PRINT:PRINT " BUT A 250KHz clock gives you 3Hz to 15.5KHz..."
580 PRINT " and very long envelope times! taking many seconds (upto 120+)
      for a sound to rise and fall"
590 PRINT@15,23,"press SPACE to continue ";
600 A=INCH:IF A<>20 THEN 590
610 CLS:PRINTA$:PRINTA$:PRINTA$:PRINT:PRINT
620 PRINT"NOW TO TRY A FEW SAMPLE SOUNDS"
630 PRINT".....0....."
      :PRINT
640 PRINT " EACH SAMPLE WILL DISPLAY THE REGISTERS USED AND THE VALUE..
      CONTAINED IN THEM!"
650 PRINT:PRINT
660 PRINT "      1-CREATE YOUR OWN SOUND      2-BOMB DROP"
670 PRINT "      3-PROP PLAIN/TRAIN              4-CHIME"
680 PRINT "      5-DRUM STROKES                    6-DUAL SOUNDS (BOTH P.S.G's)"
690 PRINT "      7-SEA WAVES                      8-REVIEW INFORMATION "
700 PRINT "      9-CALCULATE ":PRINT
710 PRINT@5,20," ENTER NUMBER OF CHOICE! ";N$=INCH$:ON VAL(N$)GOSUB730,
      1800,1990,2120,2250,2370,2510,130,3100
720 GOTO 610
730 CLS
740 PRINTA$:PRINTA$:PRINTA$
750 PRINT@3,6," ENTER '0' TO RETURN:"
760 PRINT@31,6,AB$:PRINT@31,10,AB$
770 PRINT@31,7," "
780 PRINT@31r,8," "
790 PRINT@31,9," "
800 PRINT@3,5," INPUT WHICH PSG 1 OR 2 ";PP
810 IF PP=0 THEN 610
820 IF PP<1 OR PP>2 THEN 800:P1=PP
830 IF PP=1 THEN XX=&C0:IF PP=1 THEN YY=&C1:PSG$=""
840 IF PP=2 THEN PSG$="":IF PP=2 THEN 1030
850 PRINT@26,6,"      ":PRINT@14,7," "
860 PRINT@3,7," INPUT WHICH REGISTER (0-15) ";RR
870 IF RR<0 OR RR>15 THEN 860
880 RR=RR+1
890 PRINT@1,12,D$
900 PRINT@1,15,D$:PRINT@0,18,PSG$
910 ON 7RR GOSUB2720,2740,2760,2780,2800,2820,2840,2860,2890,2910,2930,
      2950,2980,3010,3040,3070
930 RR=RR-1
940 OUTXX,RR:OUTYY,VV

```

```

950 PRINT@20,5,"      ":PRINT@26,7,"      ":PRINT@14,8,"      "
960 OUT &C0,7:N=0:PRINT@0,18,PSG$
970 IOM4,0:IOM5,0
980 FORRN=0TO4:OUT&C0,RN:S(RN)=INP(&C1):PRINT@12*RN,20," REG ";RN;
    "S(RN)" ";:NEXTRN
990 N=0:FORRN=5TO9:OUT&C0,RN:S(RN)=INP(&C1):PRINT@12*N,21," REG ";RN;
    "S(RN)" ";:N=N+1:NEXTRN
1000 N=0:FORRN=10TO13:OUT&C0,RN:S(RN)=INP(&C1):PRINT@12*N,22," REG ";RN;
    "S(RN)" ";:N=N+1:NEXTRN
1010 PRINT@18,18," ";F$=INCH$:IF F$="Y" THEN 1220
1020 IOM4,1:IOM5,1:PRINT@17,18,MUL$(" ",27):GOTO 750
1030 XX=&C2:YY=&C3
1040 PRINT@26,6,"      ":PRINT@14,7,"      "
1050 PRINT@3,7,:INPUT" WHICH REGISTER (0-15) ";RR
1060 IF RR<0 OR RR>15 THEN 860
1070 RR=RR+1
1080 PRINT@1,12,D$
1090Z PRINT@1,15,D$:PRINT@0,18,PSG$
1100 ON RR GOSUB2720,2740,2760,2780,2800,2820,2840,2860,2890,2910,2930,
    2950,2980,3010,3040,3070
1110 PRINT@3,8,:INPUT" WHAT VALUE ";VV
1120 RR=RR-1
1130 OUTXX,RR:OUTYY,VV
1140 PRINT@20,5,"      ":PRINT@26,7,"      ":PRINT@14,8,"      "
1150 OUT &C0,7:N=0:PRINT@0,18,PSG$
1160 IOM4,0:IOM5,0
1170 FORRN=0TO4:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*RN,20," REG ";RN;
    "T(RN)" ";:NEXTRN
1180 N=0:FORRN=5TO9:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*N,21," REG ";RN;
    "T(RN)" ";:N=N+1:NEXTRN
1190 N=0:FORRN=10TO13:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*N,22," REG ";RN;
    "T(RN)" ";:N=N+1:NEXTRN
1200 PRINT@18,18," ";F$=INCH$:IF F$="Y" THEN 1220
1210 IOM4,1:IOM5,1:PRINT@17,18,MUL$(" ",27):GOTO 750
1220 CLS:PRINTA$
1230 B=5:A=48:IF PP=2 THEN A=58
1240 PRINT@1,3;"REG. ";TAB(12)"DESCRIPTION";TAB(49)"PSG-1";TAB(59)"PSG-2"
1250 PRINT@0x,20;"CONTROLS:- 'W'=UP. 'Z'=DOWN. 'A'=PSG-1. 'S'=PSG-2.
    'P'=PRINT IT."
1260 PRINT@0,21;"----- 'E'=END. 'SPACE' TO ENTER VALUE."
1270 PRINT@0,22;"      'U'=RAISE VALUE BY 1. 'D'=LOWER VALUE BY 1.";
1280 RESTORE 1310
1290 FORX=0TO13:PRINT@2,X+5;X;
1300 READDA$:PRINTTAB(10)DA$;TAB(50,46)S(X);TAB(60,46)T(X);:NEXTX
1310 DATA"CH A FINE. 8 BITS","CH A COURSE. 4 BITS","CH B FINE. 8 BITS",
    "CH B COURSE. 4 BITS","CH C FINE. 8 BITS"
1320 DATA"CH C COURSE. 4 BITS","NOISE TONE. 5 BITS","ENABLE 3-BIT^S SOUND.
    3-NOISE. 2-I/O","CH A AMP","CH B AMP"
1330 DATA"CH C AMP","ENV PERIOD FINE. 8 BITS","ENV PERIOD COURSE. 8 BITS
    ","ENV SHAPE. 4 BITS"
1340 B=B+RR
1350 PRINT@A,B;">";
1360 QQ$=INCH$
1370 RA=A:RB=B
1380 IF QQ$="A"THEN A=48

```

```

1390 IF QQ$="Sr" THEN A=58
1400 IF QQ$="W" THEN B=B-1
1410 IF B<5 THEN B=5
1420 IF QQ$="Z" THEN B=B+1
1430 IF B>18 THEN B=18
1440 IF QQ$R=" " THEN GOTO 1530
1450 IF QQ$="P" THEN 3580
1460 IF QQ$="E" THEN 1680
1470 IF QQ$="U" THEN GOTO 1620
1480 IF X$="D" THEN GOTO 1620
1490 PRINT@A,RB;" "
1500 PRINT@A,B;">";
1510 GOTO 1360
1520 REM:ENTER VALUES
1530 PRINT@A,B;:INPUTVV:PRINT@A-1,B;" ";
1540 PRINT@A,B;" ":PRINT@A+1,B;VV;:PRINT@A,B;">";
1550 IF B<5 THEN B=5
1560 IF B>18 THEN B=18
1570 IF A=48 THEN XX=&C0:YY=&C1
1580 IF A=58 THEN XX=&C2:YY=&C3
1590 OUTXX,B-5:OUTYY,VV
1600 GOTO 1510
1610 REM:ALTER VALUES
1620 VV=VAL(MID$(SCRN$(B),A+2,3))
1630 IF X$="U" THEN VV=VV+1
1640 IF VV>255 THEN VV=255
1650 IF X$="D" THEN VV=VV-1
1660 IF VV<1 THEN VV=1
1670 GOTO 1540
1680 IF PP=2 THEN 1740
1690 CLS:OUT &C0,7:N=0:PRINT@0,18,PSG$
1710 FORRN=0TO4:OUT&C0,RN:S(RN)=INP(&C1):PRINT@12*RN,20," REG ";RN;
  "S(RN) " ";:NEXTRN
1720 N=0:FORRN=5TO9:OUT&C0,RN:S(RN)=BINP(&C1):PRINT@12*N,21," REG ";RN;
  "S(RN) " ";:N=N+1:NEXTRN
1730 N=0:FORRN=10TO13:OUT&C0,RN:S(RN)=INP(&C1):PRINT@12*N,22," REG ";RN;
  "S(RN) " ";:N=N+1:NEXTRN:GOTO 1790
1740 CLS:OUT &C0,7:N=0:PRINT@0,18,PSG$
1750 IOM4,0:IOM5,0
1760 FORRN=0TO4:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*RN,20," REG ";RN;
  "T(RN) " ";:NEXTRN
1770 N=0:FORRN=5TO9:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*N,21," REG ";RN;
  "T(RN) " ";:N=N+1:NEXTRN
1780 N=0:FORRN=10TO13:OUT&C2,RN:T(RN)=INP(&C3):PRINT@12*N,22," REG ";RN;
  "T(RN) " ";:N=N+1:NEXTRN
1790 PRINT@0,0;:GOTO 740
1800 CLS:PRINT:PRINTA$
1810 PRINT@15,4,"REGISTER":PRINT@43,4,"DATA"
1820 PRINT@15,5v,"~~~~~":PRINT@43,5,"~~~~~"
1830 RESTORE 1880
1840 FORX=0TO13:READA
1850 PRINT@18,X+7,X:PRINT@43,X+7,A
1860 OUT&C0,X:OUT&C1,A
1870 NEXTX
1880 DATA0,0,0,0,0,0,254,15,0,0,0,20,13

```



```
1890 FORT=1T0255:OUT&C0,0:OUT&C1,T:PRINT@42,7,">"T:NEXTT
1900 FORQ=0T013:OUT&C0,Q:OUT&C1,00:NEXTQ
1910 RESTORE 1960
1920 FORJ=0T013:READB
1930 PRINT@18,J+7,J:PRINT@43,J+7,B" "
1940 OUT&C0,J:OUT&C1,B
1950 NEXTJ
1960 DATA0,0,0,0,0,0,31,7,16,16,16,0,120,0
1970 FORW=1T01000:NEXTW
1980 RETURN
1990 CLS:PRINT:PRINTA$
2000 PRINT@15,4,"REGISTER":PRINT@43,4,"DATA"
2010 PRINT@15,5,"~~~~~":PRINT@43,5,"~~~~~"
2020 RESTORE 20q70
2030 FORX=0T013:READA
2040 PRINT@18,X+7,X:PRINT@43,X+7,A
2050 OUT&C0,X:OUT&C1,A
2060 NEXTX
2070 DATA35,0,0,0,0,0,25n,0,16,16,16,0,30,12
2080 FORX=15T01 STEP-1:OUT&C0,12:OUT&C1,X:FORT=1T01000:NEXTT:PRINT@42,19,
">";X:NEXTX
2090 FORT=1T05000:NEXT
2100 FORT=0T015:OUT&C0,T:OUT&C1,0:NEXTT
2110 RETURN
2120 CLS:PRINT:PRINTA$
2130 PRINT@15,4,"REGISTER":PRINT@43,4,"DATA"
2140 PRINT@15,5,"~~~~~":PRINT@43,5,"~~~~~"
2150 FORQ=1T010
2160 RESTORE 2210
2170 FORX=0T013:READA
2180 PRINT@18,X+7,X:PRINT@43,X+7,A
2190 OUT&C0,X:OUT&C1,A
2200 NEXTX
2210 DATA100,0,0,0,0,0,18,26,20,0,0,0,100,8
2220 FORT=1T01000:NEXTT,Q
2230 FORT=0T015:OUT&C0,T:OUT&C1,0:NEXT
2240 RETURN
2250 CLS:PRINT:PRINTA$
2260 PRINT@15,4,"REGISTER":PRINT@35,4,"DATA":PRINT@50,4,"DATA"
2270 PRINT@15,5,"~~~~~":PRINT@35,5,"~~~~~":PRINT@50,5,"~~~~~"
2280 FORQ=1T05
2290 RESTORE 2350
2300 FORX=0T013:READA:OUT&C0,X:OUT&C1,A
2310 PRINT@18,X+7,X:PRINT@35,X+7,A:NEXTX
2320 FORX=0T013:READB:OUT&C2,X:OUT&C3,B:PRINT@50,X+7,B:NEXTX
2330 FORX=0T01000:NEXTX:OUT&C0,13:OUT&C1,0:OUT&C2,13:OUT&C3,0
2340 NEXTQ
2350 DATA0,0,0,0,0,0,2,7,26,0,0,H0,30,8,0,0,0,0,0,0,7,26,0,0,0,15,8
2360 RETURN
2370 CLS:PRINT:PRINTA$
2380 PRINT@15,4,"REGISTER":PRINT@35,4,"DATA":PRINT@50,4,"DATA"
2390 PRINT@15,5,"~~~~~":PRINT@35,5,"~~~~~":PRINT@50,5,"~~~~~"
2400 RESTORE 2420
2420 DATA20,0,0,0,0,0,0,0,0,0,0,0,15,0,0,16,16,0,16,0,0,5,50,12,10
2440 OUT&C0,X:OUT&C1,A:OUT&C2,X:OUT&C3,B
```



```
2450 NEXT
2460 FORP=1TO150 STEP5:A=RND(15)+1:OUT&C0,0:OUT&C1,P*1.35:OUT&C0,12:
      OUT&C1,P:OUT&C2,6:OUT&C3,150/P:OUT&C2,12:OUT&C3,152-P
2470 OUT&C0,13:OUT&C1,A:FORQ=1TO1000:NEXTQ:PRINT@34,7,">";
      P*1.35:PRINT@34,19,">";P:PRINT@49,13,">";150/P:PRINT@49,19,">";152-P
2480 PRINT@49,20,">";A:NEXTP
2490 FORT=1TO5000:NEXTT
2500 FORT=0TO13:OUT&C0,T:OUT&C1,0:OUT&C2,T:OUT&C3,0:NEXT:RETURN
2510 CLS:PRINT:PRINTA$
2520 PRINT@15,4,"REGISTER":PRINT@43,4,"DATA"
2530 PRINT@15,5,"~~~~~";PRINT@43,5,"~~~~~"
2540 RESTORE 2600
2550 FORX=0TO13:READA
2560 PRINT@18,X+7,X:PRINT@43,X+7,A
2570 OUT&C0,X:OUT&C1,A
2580 NEXTX
2590 PRINT@1;CHR$(10);
2600 DATA0,2,0,0,0,0,15,198,16,16,6,0,65,14
2610 FORX=1TO10:A=RND(45)+35:B=RND(12)+8:C=RND(17):OUT&C0,12:OUT&C1,A:
      PRINT@42,19,">"A:OUT&C0,6:OUT&C1,B:PRINT@42,13,">"B
2620 FORT=1 TO RND(1500)+2500: NEXTT,X
2630 FORX=0TO13:OUT&C0,X:OUT&C1,0:NEXTX
2640 RETURN
2650 REM
2660 REM *****
2670 REM * SOUND DEVELOPEMENT *
2680 REM * COPYRIGHT *
2690 REM * MEL SAUNDERS 3/84 *
2700 REM *****
2720 PRINT@1,12,"R-0 8 BIT FINE TUNE CHANNEL A (PERIOD VALUE)
      HIGH VALUE LOW TONE..... AND VICE VERSA"
2730 RETURN
2740 PRINT@1,12,"R-1 4 BIT COARSE TUNE CHANNEL A (PERIOD VALUE)
      HIGH VALUE LOW TONE..... AND VICE VERSA"
2750 RETURN
2760 PRINT@1,12,"R-2 8 BIT FINE TUNE CHANNEL B (PERIOD VALUE)
      HIGH VALUE LOW TONE..... AND VICE VERSA"
2770 RETURN
2780 PRINT@1,12,"R-3 4 BIT COARSE TUNE CHANNEL B (PERIOD VALUE)
      - HIGH VALUE LOW TONE..... AND VICE VERSA"
2790 RETURN
2800 PRINT@1,12,"R-4 8 BIT FINE TUNE CHANNEL C (PERIOD VALUE)
      HIGH VALUE LOW TONE..... AND VICE VERSA"
2810 RETURN
2820 PRINT@1,12,"R-5 4 BIT COARSE TUNE CHANNEL C (PERIOD VALUE)
      HIGH VALUE LOW TONE..... AND VICE VERSA"
2830 RETURN
2840 PRINT@1,12,"R-6 5 BIT NOISE TONE VALUE (PERIOD VALUE)
      LOW VALUE HIGH TONE..... AND VICE VERSA"
2850 RETURN
2860 PRINT@1,12,"R-7 'ENABLE :note active low: BITS 0-2 CHANNELS A,B
      AND C BITS 3-5 ADD NOISE ON A,B AND C";
2870 PRINT" BITS 6-7 I/O PORT CONTROL"
2880 RETURN
2890 PRINT@1,12,"R-8 5 BIT VOLUME CONTROL FOR CHANNEL A.SETTING BIT
```

```

5 (16) GIVES      ENVELOPE CONTROL OF VOLUME"
2900 RETURN
2910 PRINT@1,12,"R-9 5 BIT VOLUME CONTROL FOR CHANNEL B.SETTING BIT
5 (16) GIVES      ENVELOPE CONTROL OF VOLUME"
2920 RETURN
2930 PRINT@1,12,"R-10 5 BIT VOLUME CONTROL FOR CHANNEL C.SETTING BIT
5 (16)
GIVES ENVELOPE CONTROL OF VOLUME"
2940 RETURN
2950 PRINT@1,12,"R-11 8 BIT FINE TUNE ENVELOPE PERIOD.AGAIN ANOTHER
PERIOD VALUE      3 HIGH VALUES GIVE LONG DECAYS ECT"
2960 PRINT@10,15,"note clock is divided by 256 and not 16"
2970 RETURN
2980 PRINT@1,12,"R-12 8 BIT COARSE TUNE ENVELOPE PERIOD.AGAIN ANOTHER
PERIOD... VALUE,HIGH VALUES GIVE VERY LONG DECAYS ECT"
2990 PRINT@10,15,"note clock is divided by 256 and not 16"
3000 RETURN
3010 PRINT@1,12,"R-13 4 BIT ENVELOPE SHAPE/CYCLE BIT-0 HOLD.BIT-1
ALTERNATE.        BIT-2 ATTACK.BIT-3 CONTINUE."
3020 PRINT@1,14," NOTE- bits can be set in combination to provide
various           waveforms triangular,square,ect."
3030 RETURN
3040 PRINT@1,12,"R-14 8 BIT I/O PORT-A BIT 6 OF REG-7 MUST BE SET LOW
FOR INPUT        AND VISA VERSA"
3050 PRINT@10,15,"note all bits are held high on input
                 (and must be taken low to enter value!)"
3060 RETURN
3070 PRINT@1,12,"R-15 8 BIT I/O PORT-B.BIT 7 OF REG-7 MUST BE SET LOW
FOR INPUT        AND VISE VERSA"
3080 PRINT@10,15,"note all bits are held high on input
                 (and must be taken low to enter value!)"
3090 RETURN
3100 CLS:PRINTA1$:PRINTA$:PRINTA1$:PRINT
3110 REM
3120 PRINT@0,5,"Do you want to calculate:":PRINT@0,11,D$:PRINT@0,12,
D$:PRINT@0,13
3130 PRINT@5,6,"TONE FREQ, REGISTER VALUES or ENVELOPE (T/R or E) ";
:AA$=INCH$
G3140 IF AA$="R" THEN 3280
3150 IF AA$="E" THEN 3360
3160 PRINT@3,8:INPUT"SYSTEM CLOCK FREQ (Hz) ";ST
3170 PRINT@3,9:INPUT"COARSE REGISTER VALUE (0-15) ";CT
3180 IF CT>15 THEN 3260
3190 PRINT@3,10:INPUT" FINE REGISTER VALUE (0-255) ";FT
3200 IF FT>255 THEN 3260
3210 TP=(CT*256+FT)*16:IF TP>65535 GOTO 3260
3220 FR=ST/TP:IF FR<1000 THEN FR$="TONE="+STR$(FR)+" Hz":GOTO 3240
3230 IF FR=>1000 THEN FR$="TONE="+STR$(FR/1000)+" KHz"
3240 PRINT@10,12,FR$
3250 GOTO 3270
3260 PRINT@10,12,"3"
3270 GOTO 3350
3280 PRINT@3,8:INPUT"SYSTEM CLOCK FREQ (Hz) ";ST
3290 PRINT@3,9:INPUT"REQUIRED TONE FREQ (Hz) ";FF
3300 C=ST/16/FF:CT=C/256:FT=C MOD 256
3310 IF CT>15 THEN 3340

```

```

3320 PRINT@10,11,"COARSE REG="INT(CT):PRINT@10,13,"FINE REG="FT
3330 GOTO 3350
3340 PRINT@10,12,""
3350 PRIINT@5,22,"PRESS 'C' to continue calculations or others to quit ";
:AN$=INCH$:IF AN$="C" THEN 3100:ELSE RETURN
3360 PRINT@0,6,D$:PRINT@5,6,"ENVELOPE PERIOD TIMES or REGISTER VALUES
(P/R) ";AQ$=INCH$
3370 IF AQ$="R" THEN 3500
3380 PRINT@3,8:INPUT"SYSTEM CLOCK FREQ (Hz) ";ST
3390 PRINT@3,9:INPUT"COARSE REGISTER VALUE (0-255) ";CT
3400 IF CT>255 THEN 3480
3410 PRINT@3,10:INKPUT" FINE REGISTER VALUE (0-255) ";FT
3420 IF FT>255 THEN 3480
3430 TP=(CT*256+FT)*256:D=ST/TP
3440 IF D<1 THEN FR$="PERIOD~D="+STR$(1/D)+" Seconds":GOTO3470
3450 IF D>1 AND D=<1000 THEN FR$="PERIOD="+STR$((1/D)*1000)+"
Milliseconds":GOTO 3470
3460 IF D>1000 THEN FR$="PERIOD="+STR$((1/D)*1000000)+" Microseconds"
3470 PRINT@10,12,FR$:GOTO 3490
3480 PRINT@10,12,"':":GOTO 3350
3490 GOTO 3350
3500 PRINT@3,8:INPUT"SYSTEM CLOCK FREQ (Hz) ";ST
3510 PRINT@3,9:INPUT"ENVELOPE PERIOD (SECS) ";FF
3520 C=ST/(256/FF):CT=C/256:FT=C MOD 256
3530 IF CT>255 OR FT>255 THEN 3560
3540 PRINT@10,11,"COARSE RE.G="INT(CT):PRINT@10,13,"FINE REG="FT
3550 GOTO 3350
3560 PRINT@6,12,""
3570 GOTO 3350
3580 PRINT@3,23:INPUT" ";TT$
3590 PRINT@1;CHR$(27);CHR$(64);A$
3610 NT=((LEN(TT$)+7)/2):NNT=32-NT
3620 PRINT@1;TAB(NNT,32);"TITLE=" +TT$
3630 PRINT@1;CHR$(10);
3640 PRINT@1;TAB(3);"REG.";TAB(12)"DESCRIPTION";TAB(49)"PSG-1";TAB(59)"PSG-2"
3650 PRINT@1;CHR$(10);
3660 RESTORE 3690
3670 FORX=0TO13
3680 READDA$:PRINT@1;TAB(4,32);X;TAB(10)DA$;TAB(50,46)S(X);TAB(60,46)T(X):
NEXTX
3690 DATA"CH A FINE. 8 BITS","CH A COURSE. 4 BITS","CH> B FINE.
8 BITS","CH B COURSE. 4 BITS","CH C FINE. 8 BITS"
3700 DATA"CH C COURSE. 4 BITS","NOISE TONE. 5 BITS","ENABLE 3-BITS JSOUND.
3-NOISE. 2-I/O","CH A AMP","CH B AMP"
3710 DATA"CH C AMP","ENV PERIOD FINE. 8 BITS","ENV PERIOD COURSE. 8 BITS",
"ENV SHAPE. 4 BITS"
3720 PRINT@0
3730 GOTO 1680

```

DISK SOFTWARE

The newsletter will list any disk software that members may wish to make available to others. It should run under CP/M 2.2 on a normal Interak. Please note that the supplier is liable for bug repairs, delivery and assistance.

CP/M 2.2

The industry standard disk operating system for the Interak computer. Needs a VDU 2K. Customised BIOS by Wolf Schroeder with customised MOVCPM. Supplied with multi diameter diskette Formatter and considerable extra utilities not normally provided by CP/M implementations. Ready to Boot up and go on a 3.5" disk.

Contact Greenbank Electronics for further details.

ZYBASIC 4

Runs to CP/M 2.2 with a VDU 2K. Any standard BIOS. Supports CP/M disk file program storage. Enhanced Keyword set, memory manager to allow correct use of available memory. 11 characters long variable names, Multi-dimensioned arrays of up to 32767 dims each up to 32767. Fully downward compatible with previous versions and will load programs from tape or disk and save them to tape or disk. Supplied on a 3.5" disk with run time program source files. New version of the manual is available seperatly. Previous users can be upgraded at cost.

Contact Greenbank Electronics for further details.

TAPE SOFTWARE

See CONTACTS page at the end of this issue for "ORDER FROM" addresses. Software supplied is the responsibility of the "ORDER FROM". Please deal directly with the "ORDER FROM" in the event of bugs ect.

You may use this section to sell software to other users. Send a brief description of your product giving details of its distribution and price, to the EDITOR. Note that you will be responsible for the support of your own product. The newsletter cannot be held responsible for or get involved with duff code or distributors. Of course we will publish letters deriding any product that fails to live up to its claims.

MACHINE CODE

NAME	DESCRIPTION	VDU	ORDER FROM	COST
FIGFORTH	FORTH COMPILER	2K	P.VELLA	£15.00
INTERPLAY	BULLETIN BOARD DRIVER	2K	M&M ELECTRONICS	£ 4.00
MEGABUG	DEBUG/TRAINING PACKAGE	2K	P.VELLA	£13.00
VELTEXT	TEXT EDITOR	2K	P.VELLA	£ 5.00
XTAL BASIC	14K BASIC	2K	P.VELLA	£40.00
ZYBASIC 3A	INTERAK BASIC (TAPE)	2K	GREENBANK	£15.95
ZYBASIC 3C	INTERAK BASIC (ROM)	2K	GREENBANK	£27.75
ZYMON 2.V203	INTERAK MONITOR	2K	GREENBANK	£15.95

=====

XTAL BASIC

NAME	DESCRIPTION	VDU	ORDER FROM	COST
AWARI	GAME	2K	M.SAUNDERS	PP
BIORYTHMS		2K	M.SAUNDERS	PP
CHAR DES	CHARACTER DESIGNER	2K	M.SAUNDERS	£ 5.50
I-SPY	GAME	2K	M.SAUNDERS	PP
SOUND DEV	SOUND DEVELOPMENT	2K	M.SAUNDERS	£ 5.50

=====

Key: PP = Postage & packing.
POA = Please enquire (Phone for price.)

SMALL ADS

FOR SALE

Tape Labels 3"x0.675" in red, blue, green and yellow	£ 2.00p per 100
Tape/Disk labels 4"x1.5" in blue, green and yellow	£ 3.00p per 100
Disk labels 2.75"x1.5" for 3.5" drives pack of 20x5 colours ..	£ 4.00p per 100
Printout ruler/Template 16"x3.5" smoked grey, with clear lined window for reading printout, 23 flowchart cutouts.....	£ 4.70p each
Disk boxes for 10 x 3.5"	£ 4.50p each
Disk boxes for 10 x 5.25"	£ 6.20p each
Disk binder for 3.5"	£10.40p each
Disk binder for 5.25"	£13.25p each
Disk binder for 8"	£16.60p each
Epsom printer ribbons (80 series)	£ 2.76p each
Epsom printer ribbons (100 series)	£ 4.00p each
Printout binder 9.785"x12" burst/unburst	£ 3.75p each
Printout binder 17.5"x11" burst/unburst	£ 4.25p each

All require postage to be included.

MEL SAUNDERS, 7 DRUMCLIFF RD, THURNBY LODGE, LEICESTER, LE5 2LH.

CONTACTS

- CONTACT TAPES. Communicate with other members by cassette tape.
Point Contact tapes, 7 Drumcliff Rd, Thurnby Lodge, Leicester, LE5 2LH.
- BACK ISSUES... Can be obtained from:-
D.Parkins, Greenbank Electronics, 92 New Chester road, New Ferry, Wirral, Merseyside, L62 5AG.
- BOOKS..... Lend, borrow, and swap books via :-
R.E.Bowyer, 45 Ford drive, Yarnfield, Stone, Staffs.
- DATA SHEETS... Swap, borrow, lend, chip data sheets
7 Drumcliff road, Thurnby Lodge, Leicester, LE5 2LH.
- EDITOR..... Send submissions to :-
R.Eldridge, 28 Wycherley Close, Blackheath, London, SE3 7QH.
- GREENBANK Greenbank Electronics Ltd, 92 New Chester road, New Ferry, Wirral, Merseyside, L62 5AG.
- M&M ELECTRONICS, 8 Ayre view, Bride, Isle of man.
- M.SAUNDERS ... M.Saunders, 7 Drumcliff road, Thurnby Lodge, Leicester, LE5 2LH.
- MEMBERSHIP.... To join, renew or change your details contact :-
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.
- P.VELLA 19 Ford Drive, Yarnfield, Staffs.
- R.ELDRIDGE ... 28 Wycherley Close, Blackheath, London, SE3 7QH.
- SUBSCRIPTIONS. For information and payments please contact :-
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.